



Education

# **Find and Select the Right File Storage for your Applications**

Philippe Nicolas, KerStor

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - ◆ Any slide or slides used must be reproduced in their entirety without modification
  - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

**NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.**

- **Title: Find and Select the Right File Storage for your Applications**
- **Abstract:** Many businesses are linked to file storage technologies as many of these new, recent and even existing applications morph now, rely and support file based data. At the same time, the volume of data explodes especially the file data type which now represents by far the larger portion of enterprise data. With the complexity and variety of market solutions, the challenge for IT buyers and storage managers is to choose and adopt the most adapted solutions aligned to their business and IT needs to address their current and future challenges with a special attention to compliance and data retention regulations, Backup and Archiving, ILM and Tiering. This session covers the most common deployed applications, their attributes in term of file storage needs and maps these to file storage solutions available in the industry with technologies details and advantages. Various technologies are presented in this session, among them: Clustered, SAN-based, Distributed, Parallel File Storage and the very last approach Cloud Storage.
- **Learning objectives:**
  1. With a top-down approach, this tutorial improves file storage technologies positioning and understanding aligned to applications needs and challenges.
  2. The presented survey, technologies segmentation and features matrix, helps end-users, IT and file storage buyers to select, choose and adopt the right solution.
  3. And finally this content contributes to the promotion of file storage technologies in an agnostic philosophy.
- **Audience:** IT & Storage Architect, IT Manager and Buyers.

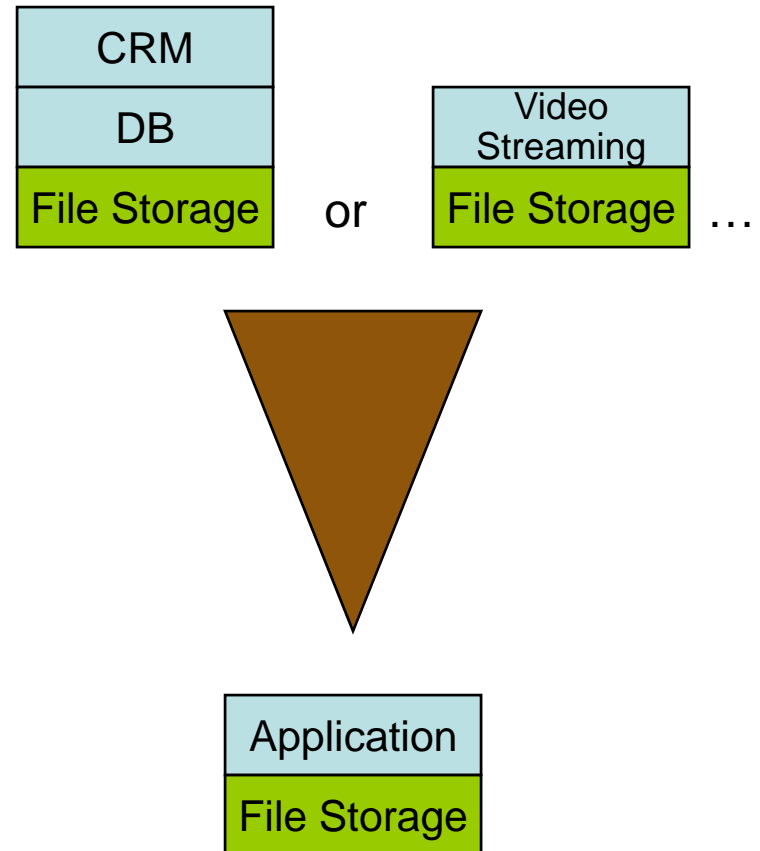
- Applications
  - ◆ Convention, Needs & Challenges
  - ◆ Type, Characteristics, I/O Patterns & Access
- The Right File Storage for Each Application
  - ◆ Application attributes and File Storage details
  - ◆ Basic & Advanced File Services
- Conclusion

# Applications

**Convention, Needs & Challenges**  
**Type, Characteristics, I/O Patterns & Access**

## ➤ Convention

- ◆ No distinction between software services above File System/Storage logic & layer
- ◆ Everything is an Application: Database, Web Server, Computing (HPC), Backup, Archival & ILM, Video streaming... even multi-tiers or multi-layers
- ◆ **Remark - Exclusion:** Application can run on many various *local disk file systems* in a single server-storage domain but this “classical” approach is not covered in this tutorial



# End-User Needs and Examples of use

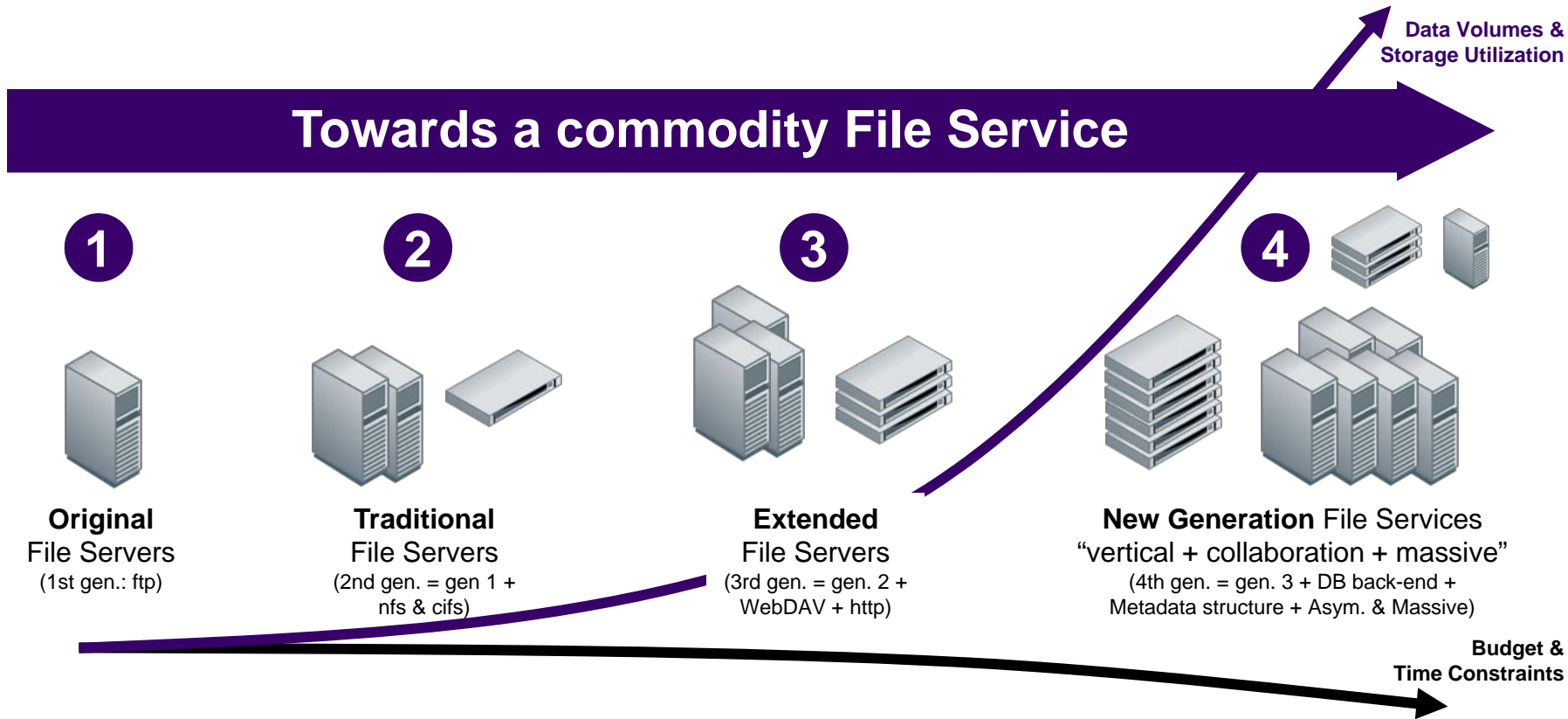
## End User Needs

- ◆ **Better scalability**
  - ◆ Capacity: fast growth of volume of data, # files, filesize...
  - ◆ Performance: IOPS, BW, frame/sec...
  - ◆ Data Sharing: avoid data duplication, aggregate more servers
  - ◆ Larger server can be expensive
- ◆ **More Availability, no Downtime**
  - ◆ Local (clustering, failover...)
  - ◆ Remote (wide failover + data replication...)
  - ◆ Global (multi-sites)
- ◆ **Easy Manageability and Administration**
  - ◆ Migration and Consolidation (data movement)
  - ◆ Remote site and file server management
- ◆ **Industry Standard**
  - ◆ Protocols, components, COTS...
- ◆ **Advanced features**
  - ◆ Load Balancing, Quotas, Security, Data Protection (snapshot, replication...), ILM & Classification, Data Reduction, Encryption, Content Indexing & Search, Reporting & Statistics, XAM...
- ◆ **Cost Reduction**
  - ◆ \$/TB, \$/IOPS, \$/BW, \$/Transac fs op., \$/NFSops...

## Examples of use

- ◆ **High Availability Clusters (local & geographic)**
  - ◆ Multiple point of presence
- ◆ **Scaling applications**
  - ◆ Web Servers - Read mostly/load balanced
  - ◆ Databases/OLTP/DW - Mostly use direct I/O
- ◆ **Distributed and Parallel app. and fast failover**
  - ◆ Data acquisition and « demanding » computing
- ◆ **Systems, App. and Data Consolidation/Migration, ILM**
  - ◆ Tech. Refresh
  - ◆ Obsolescence
- ◆ **Off-host processing**
  - ◆ Based on shared file system
  - ◆ Can also use by Point-in-Time copy techniques (not related to our data sharing definition)
- ◆ **Cloud Storage**
  - ◆ Online Backup, Data Archiving, Capture and Indexing, DR and new tier of storage consideration

# Evolution of File Services



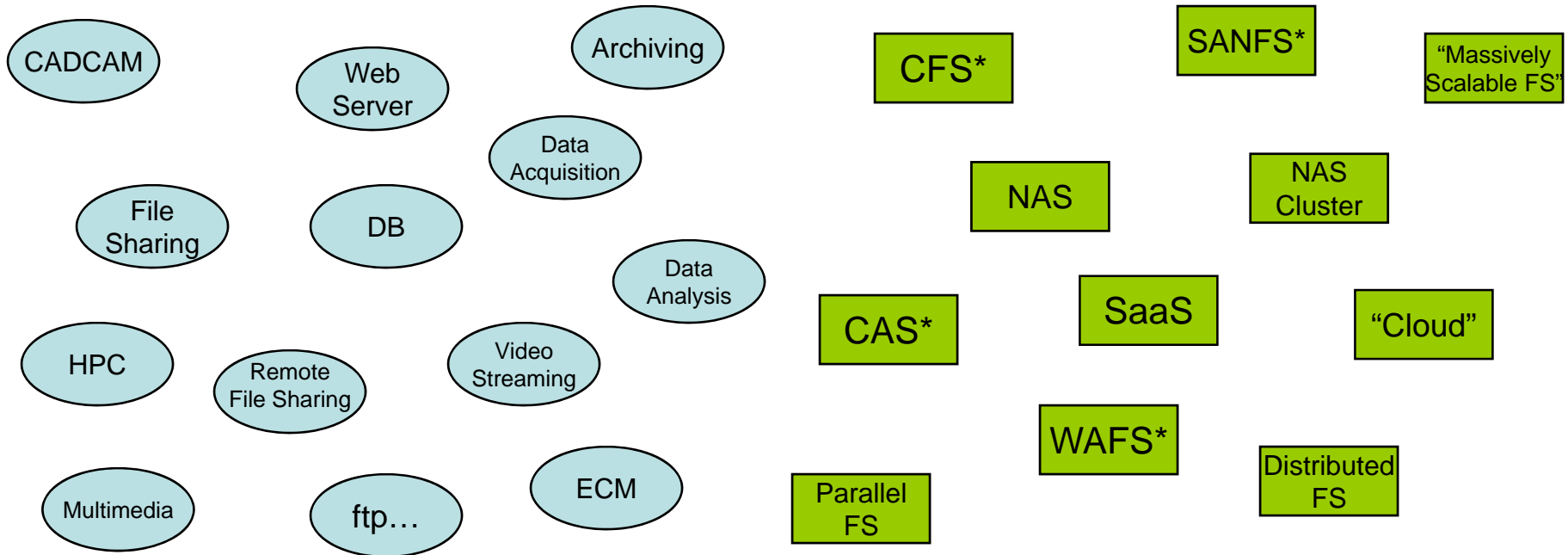
## Average disk allocation

- ▶ for Windows systems: 25-40%
- ▶ for Unix/Linux systems: 30-45%
- ▶ for iSeries and z/OS : 60-80%

## File Server/NAS considered as NAS Target

- ▶ for Backup (with DeDup) & Archiving (WORM|CAS)
- ▶ for Virtual Servers/Machines images
- ▶ for Vertical, Collaboration, HPC, Video, Cloud...

# Applications & File Storage



➤ Based on Applications characteristics and I/O behaviors, the goal is to select the right File Storage technologies among many, many existing approaches

\* CFS: Cluster File System | SANFS: SAN File System | WAFS: Wide-Area File Service (WAN Optimization & Acceleration) | CAS: Content Addressable Storage

# Applications characteristics

<b>Workload profile</b>	<b>OLTP</b>	<b>Small Data Mart</b>	<b>Home Directory</b>	<b>Large Scale Streaming (web farm)</b>	<b>High-Frequency Meta-Data update (small file create/delete)</b>
<b>Latency sensitive</b>	High	Med	Low	Low	High
<b>Throughput</b>	High R/W	High read	Low	High read	High write
<b>Concurrent sharing</b>	High	High	Low	High read	Low
<b>Caching (re-read rate)</b>	High	High	High	Low	Low

# File Storage usage

	Processor Farms	Office Env.	Archive Data	Streaming Media	Database	Web and Content Management
Typical applications	Financial simulation, Grid computing, Asic Simulation, Oil & Gas Applications, Rich Digital Content creation – Rendering	Spreadsheet, Word processing, Presentation, Pictures editing...	Medical records & imaging, insurance policy stores, gov. records, check storage applications, video surveillance	Online content delivery Direct Pay per View News distribution online radio audio streaming social networking	Exchange, SQLServer, Oracle, other OLTP...	Web farms IIS, Apache, CAD, SW dev.
Usage of File Storage	Used as a Virtual memory	Used as a shared storage pool	Used as long retention storage for online archive of records, documents and images	Used as a large and performance storage pool	Used as alternative to “traditional” raw and local disk file system	Used as a generic repository
I/O patterns & access	Equal mix of reads and writes  Sequential and large transfer sizes	Random reads and small request sizes	Large sequential access	Mixed small and large sequential transfer	Mix read/write small size	Small random reads

# **The Right File Storage for Each Application**

**Application attributes and File Storage details  
Basic & Advanced File Services**

# Some approaches

## ➤ Top-Down

- ◆ Understand, monitor and profile application I/O pattern
- ◆ Configure network stack, volumes, LUN..., stripe size and file system (block size...)
- ◆ Tune, verify & control, adapt...

## ➤ Bottom-Up

- ◆ Configure network stack, storage, LUN and volumes, stripes + file system
- ◆ Build and align application I/O size to the above one (need source)
- ◆ Tune, verify & control, adapt...

## ➤ Other

- ◆ Too many various I/O access and behaviors
- ◆ Storage and Application modification not possible

## ➤ Application

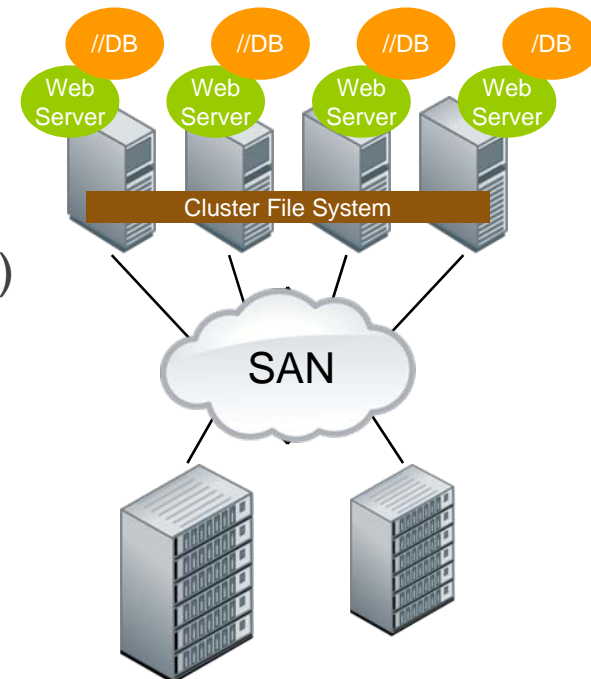
- ◆ Web Server farm with http, ftp... services
  - > Characteristics: Thousands/millions concurrent IOPS read oriented, few updates, multiple data/file format, potential DB addition
  - > Needs: Medium to High Read throughput, performance scalability (linear)
  - > Options: Load balancer in front...
- ◆ Parallel database
  - > Characteristics: Concurrent access (1 physical DB accessed by multiple instances), thousands/millions IOPS, latency influence
  - > Needs: High R/W throughput, performance scalability (linear), integrated failover
  - > Options: snapshot, HA/BC/DR...

## ➤ Configuration

- ◆ Cluster File System
  - > From 2 to 16/32 nodes

# Cluster File System

- **Cluster File System (CFS)**, also named **Shared Data Cluster**
- A Cluster FS allows a FS and files to be shared
- Centralized (asymmetric) & Distributed (symmetric) implementation
  - ◆ Centralized uses master node for meta-data updates, logging, locking...
- All nodes understand Physical (on-disk) FS structure
  - ◆ The FS is mounted by all the nodes
  - ◆ Single FS Image (Cache Coherence)
- **Lock Mechanism**
  - ◆ Distributed or Global Lock Management (DLM/GLM)



# Multimedia application

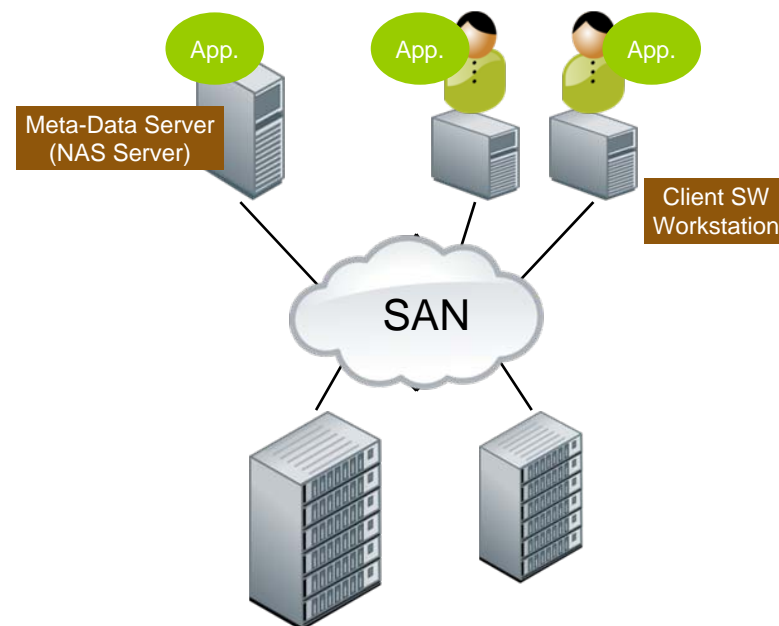
## ➤ Application

- ◆ Audio/Video Streaming, Clip Editing, Movie rendering...
  - Characteristics: Hundreds/thousands concurrent sequential BW operations (acquisition, edition...), multiple data/file format
  - Needs: Medium to High RW throughput, performance scalability (linear) for servers, hundreds of clients tolerance, thousands of application consumers, open protocols, heterogeneous OS
  - Options: Tiering, Classification, Capacity Optimization, Indexing...

## ➤ Configuration

- ◆ SAN File System

- **SAN File System (SAN FS) aka SAN File Sharing System**
- A SAN FS allows files to be shared
- Client/Server or Master/Slave (aka asymmetric) model
  - ◆ Mixed role between direct data access with host based thin software and NAS access
- Flexibility of network FS at SAN speed
- Designed to support hundreds or thousands of nodes
- Lock Mechanism & Cache Coherency



# CFS vs. SAN FS

Characteristics & Features	Cluster FS	SAN FS
Shared elements	File System and files	Files only
# of nodes	Dozens	Hundreds
Heterogeneous OS	No	Yes
Tolerance of Distance (between server and clients)	Limited	Important
Dedicated Meta-Data Server(s)* required	No (except centralized design or configuration)	Yes, usually
Physical File System layout knowledge	All nodes (Cluster FS currently requires same OS)	Meta-data server only (clients may understand if same OS)
Single File System Image (SFSI)/Cache coherency	Yes	No
File Sharing behavior	Exportable by all nodes in the cluster	No file sharing by clients File sharing by meta-data server especially if NFS/CIFS based

\* Meta-Data Server aka Master Server

# Data Acquisition, File Sharing

## ➤ Application

- ◆ Data Acquisition
  - > Characteristics: Hundreds/thousands concurrent sequential BW operations, multiple data/file format
  - > Needs: Medium to High RW throughput, performance scalability (linear) for servers, hundreds of clients tolerance, open protocols
- ◆ File Serving/Sharing, File Storage Consolidation, Data/File repository, Office application... even remote file access
  - > Characteristics: Hundreds/thousands concurrent IOPS, multiple data/file format
  - > Needs: Performance scalability, hundreds/thousands of clients/applications tolerance, embedded data protection, directory integration, open protocols
- ◆ DataBase on NFS
  - > Characteristics: Hundreds/thousands concurrent IOPS
  - > Needs: Performance (DIO, AIO...) + scalability, optimized failover, embedded data protection, open protocols

## ➤ Configuration

- ◆ Distributed approach with Network File System, aggregation of file servers (FAN, NFV/NFM) and WAFS/WAAS/WADS with SMB2

# Network File Server aka NAS

## ➤ Distributed File System – General Characteristics

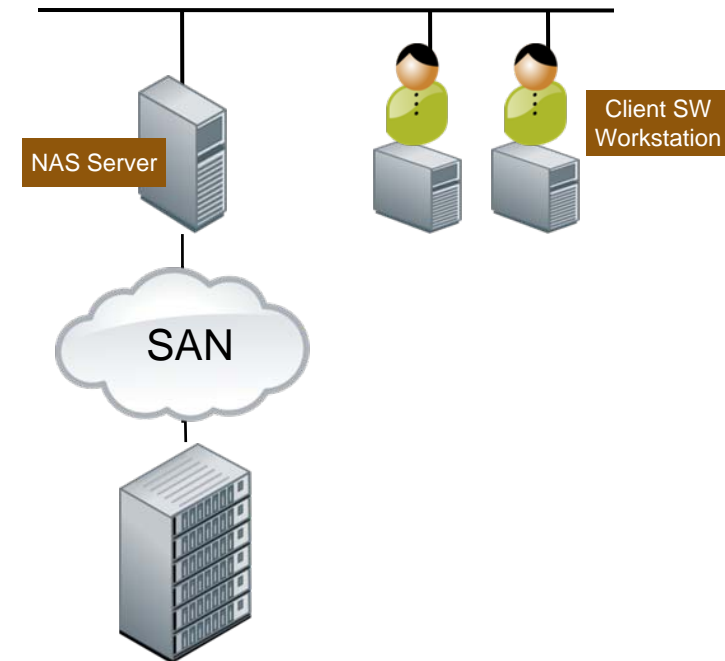
- Network transparency, User Mobility, Fault Tolerance, Scalability, File Mobility
- No File Server aggregation by default

## ➤ NFS primarily for Unix and CIFS for Windows (NAS protocols)

- Asymmetric (Client/Server) architecture
- Uses TCP/IP (UDP for NFS in the past, NFS over RDMA)
- De facto standards today
- Too “chatty”/verbose for remotes file access

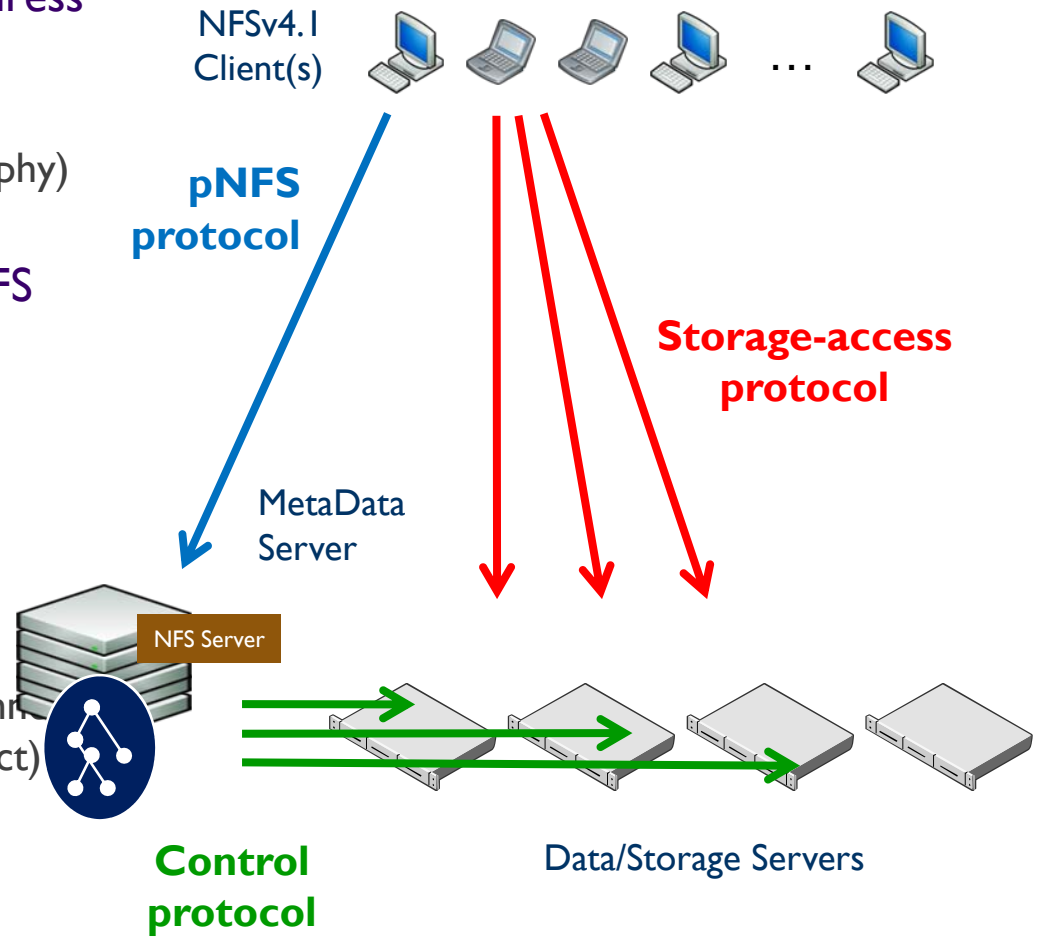
## ➤ Evolution (recent NAS protocols development)

- SMB/CIFS v1 to v2 «WAN optimized» (v 2.002): >30 times **faster** compared to SMB1 over WAN and 2-10x on LAN, **compound mechanism** (aggregation of multiple requests – only 19 commands, reduce round trips), “**durable file handles**”, **larger buffer sizes**, sym. links, secure and robust...
- NFS v2 to 3 and now v4 (v4.1): **Only 1 port** (2049), **stateful, compound operations, client caching + delegation**, security (authentication + Windows ACLs), migr. + repl., Unix/Linux & Window support, RDMA, TCP (only), **Namespace ext. (Mirror mounts & referral) + FedFS + pNFS (v4.1)**...



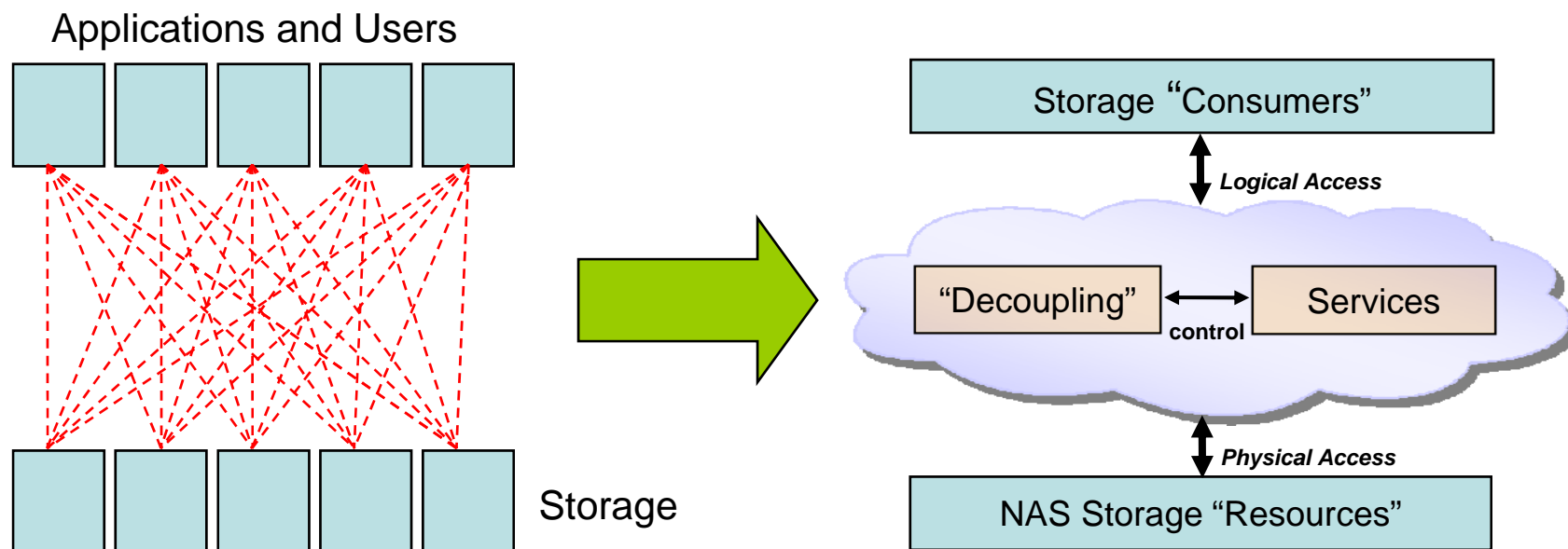
# Parallel NFS with NFSv4.1 (pNFS)

- pNFS is about scaling NFS and address file server bottleneck
  - ◆ Same philosophy as SAN FS (master/slave asymmetric philosophy) and data access in parallel
- Allow NFSv4.1 client to bypass NFS server
  - ◆ No application changes, similar management model
- pNFS extensions to NFSv4 communicate data location to clients
  - ◆ Clients access data via Fibre Channel, FCoE & iSCSI (block), OSD (object) or NFS (file)
- [www.pnfs.com](http://www.pnfs.com)

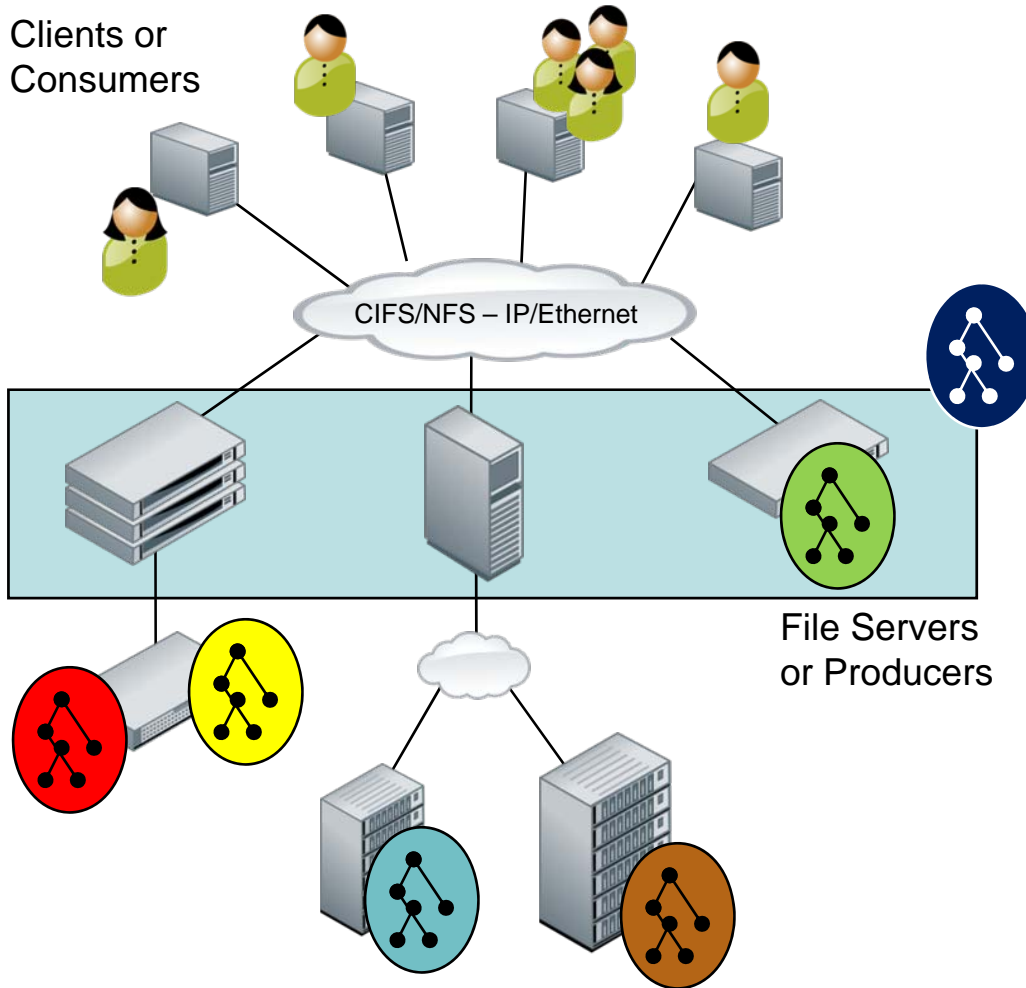


## ➤ Definition (SNIA Dictionary)

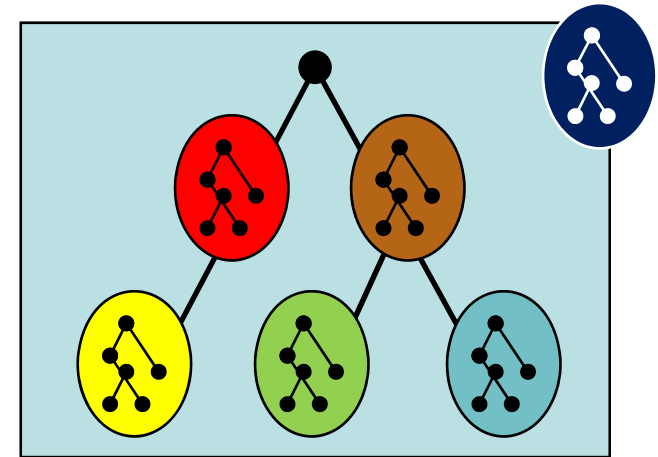
- ◆ “A **namespace-based network-oriented infrastructure** for files that includes a **decoupling layer** which **separates logical file access from physical file location**. This decoupling layer enables a **variety of services** (e.g., replication and migration) to be applied to files and filesystems”



# Namespace concept

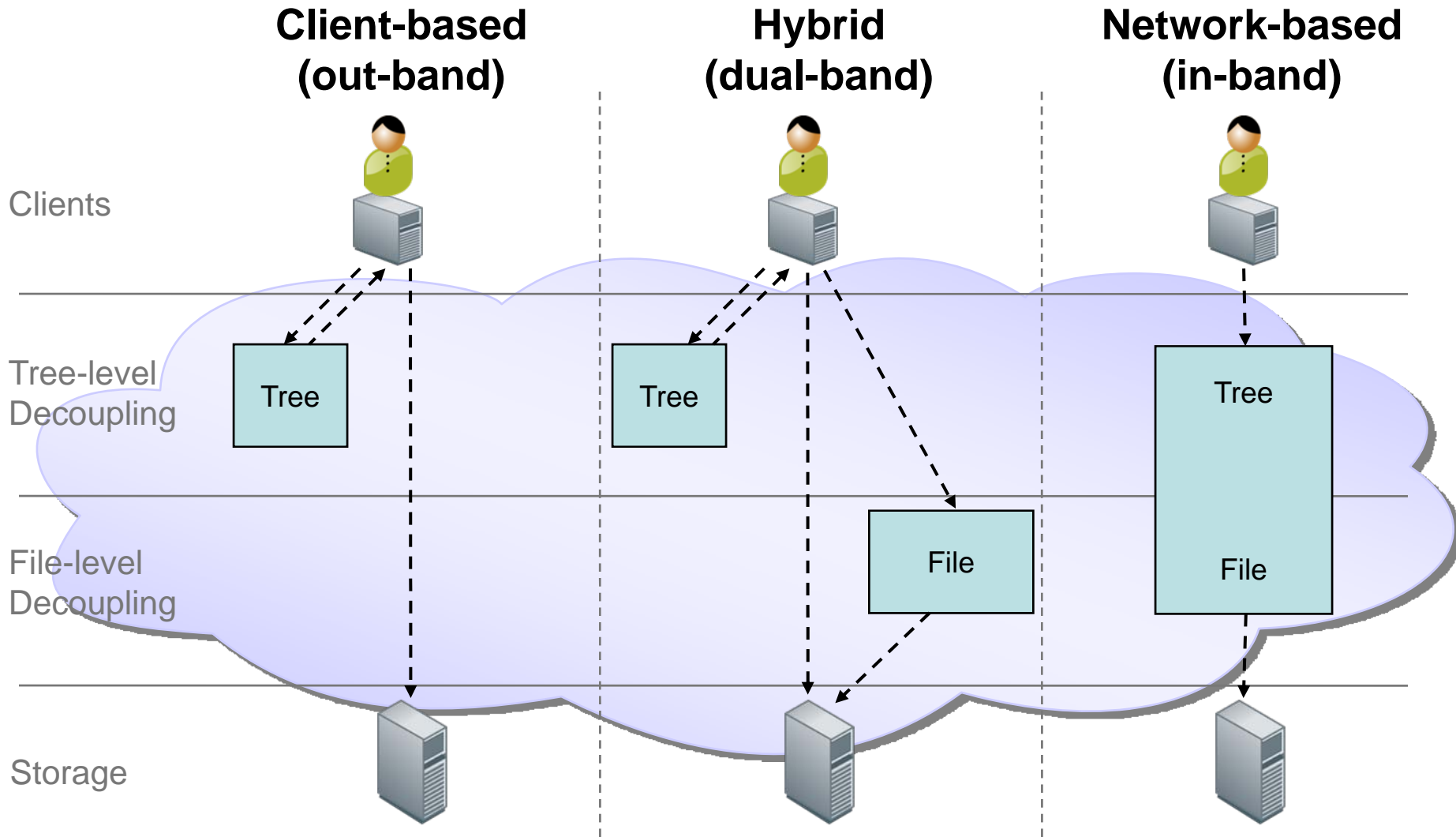


## Namespace Aggregation



- ◆ **Shared Namespace**
  - ◆ Proprietary approach
  - ◆ “internal” aggregation of same brand/model file/storage servers
- ◆ **Global Namespace**
  - ◆ Open approach
  - ◆ “external” aggregation of individual file servers + shared namespace if any

# Decoupling Approaches



## ➤ File Virtualization

- ◆ Capacity to mask physical location across (file) servers and provide logical access among them
- ◆ Seen as one logical entity
- ◆ No network, NAS protocols or client (consumer) related
- ◆ Example : Cluster File System...

## ➤ Network File Virtualization (NFV)

- ◆ Integration of network, NAS protocols and clients (consumers) on top of (file) servers (notion of FAN)
- ◆ Example : DFS, Automount, NFS v4 namespace extensions (mirror mounts, referral & FedFS)...

## ➤ File Management

- ◆ = File Virtualization + File Services
  - File Services: Replication, Failover, Load Balancing, Quotas, Security, Data Protection (snapshot, replication...), ILM & Classification, Data Reduction, Encryption, Content Indexing & Search, Reporting & Statistics, XAM...

## ➤ Network File Management (NFM)

- ◆ = NFV + File Services

## ➤ Application

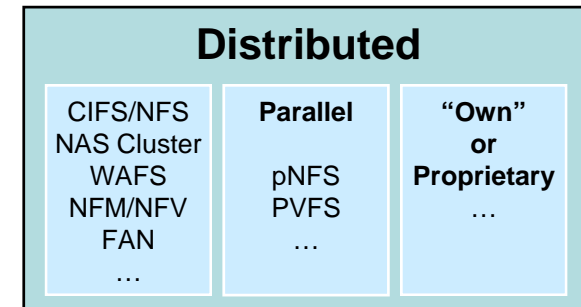
- ◆ Data intensive application, Data Collection/Logging, Data Analysis, High Performance Computing, High “stress” application...
  - Characteristics: Hundreds/thousands concurrent BW & IOPS, large data transfer, multiple data/file format
  - Needs: Very High RW throughput, logical consolidation, performance scalability (linear) for servers (1000s), 100s to 10000s of clients tolerance, open protocols, 10-100-100PBs of capacity, full embedded resiliency and redundancy, global, permanent and ubiquity access and presence, notion of global/shared namespace, standard or proprietary file access, security and privacy
  - Options: Load balancing, Tiering, Storage Capacity Optimization...

## ➤ Configuration

- ◆ Distributed File System (back-end data distribution and parallel)

# Distributed implementations

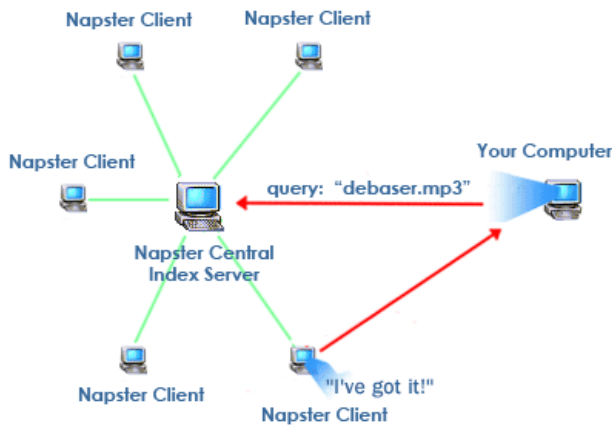
- “Aggregation | Union” of storage servers
- Symmetric, Asymmetric or P2P
  - ◆ Central Authority aka Master node or Masterless
- Parallel vs. non-Parallel
  - ◆ File striped and concurrent access, single node access
- Shared-nothing model
  - ◆ Aggregation of file storage/servers
- User-mode vs. Kernel/System-mode
- File-based vs. object-based
- Notion of Shared/Private Namespace (storage nodes wide)
- RAIN, Grid implementation with embedded data protection and resiliency
  - ◆ No storage nodes are fully trusted and are expected to fail at any time
- Extended features
  - ◆ Policies enforcement for Automation, Snapshot, CDP, Replication, Mirroring, ILM/FLM/Tiering, Migration, Load balancing...



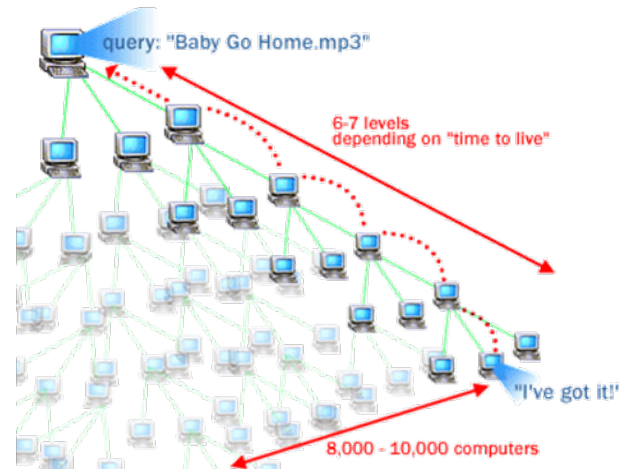
# P2P Implementations

➤ Interesting implementation with aggregation of other machines storage space with or wo a central server (asym or sym P2P)

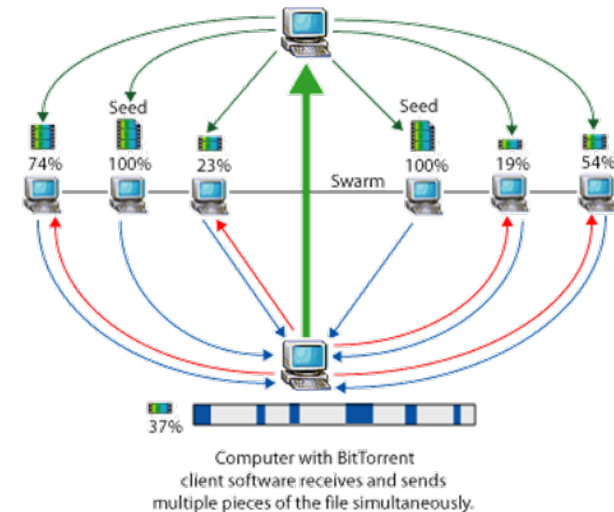
- ◆ Ex: P2P File Sharing System such as music, mp3...



Napster



Gnutella

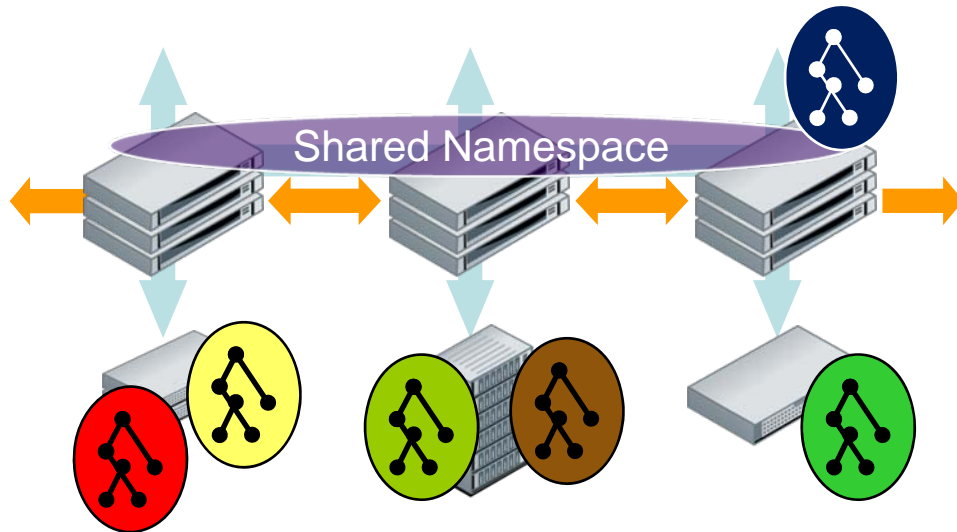


BitTorrent

## Symmetric Philosophy

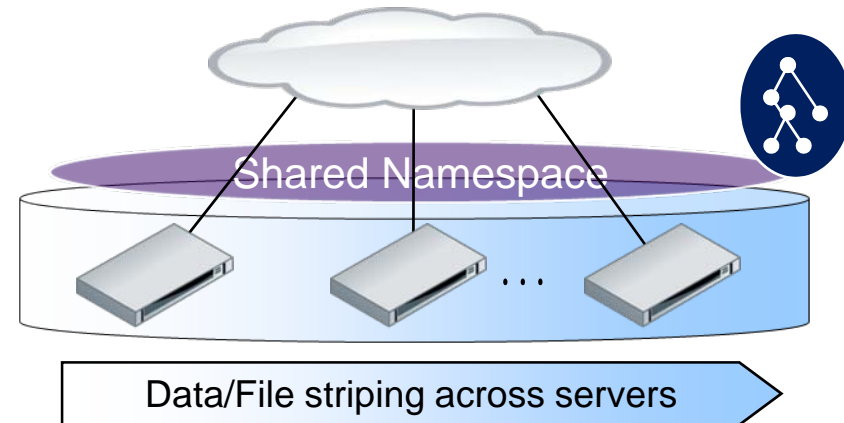
### ➤ Aggregation of **independent** storage servers

- ◆ File entirely stored by 1 storage server (no file striping)

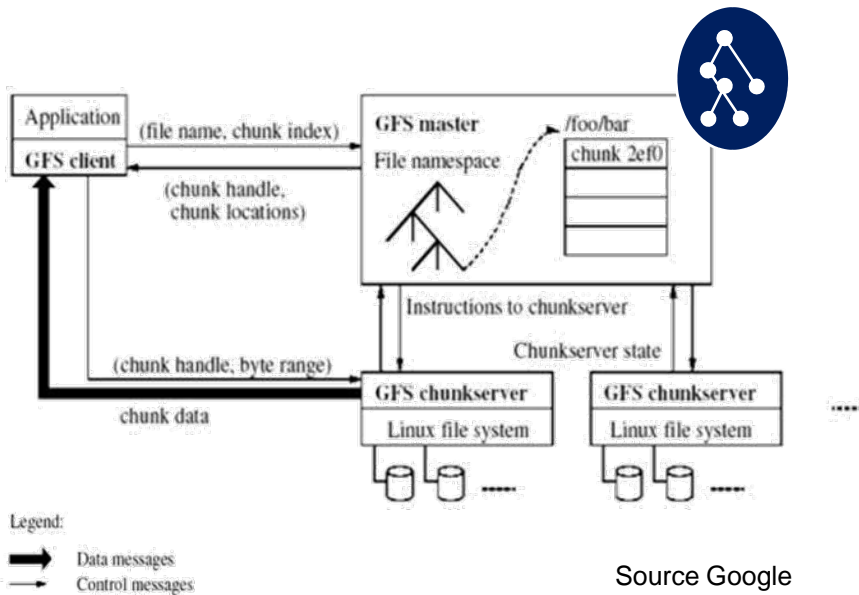


### ➤ Aggregation of **homogeneous** storage servers

- ◆ File is striped across storage server

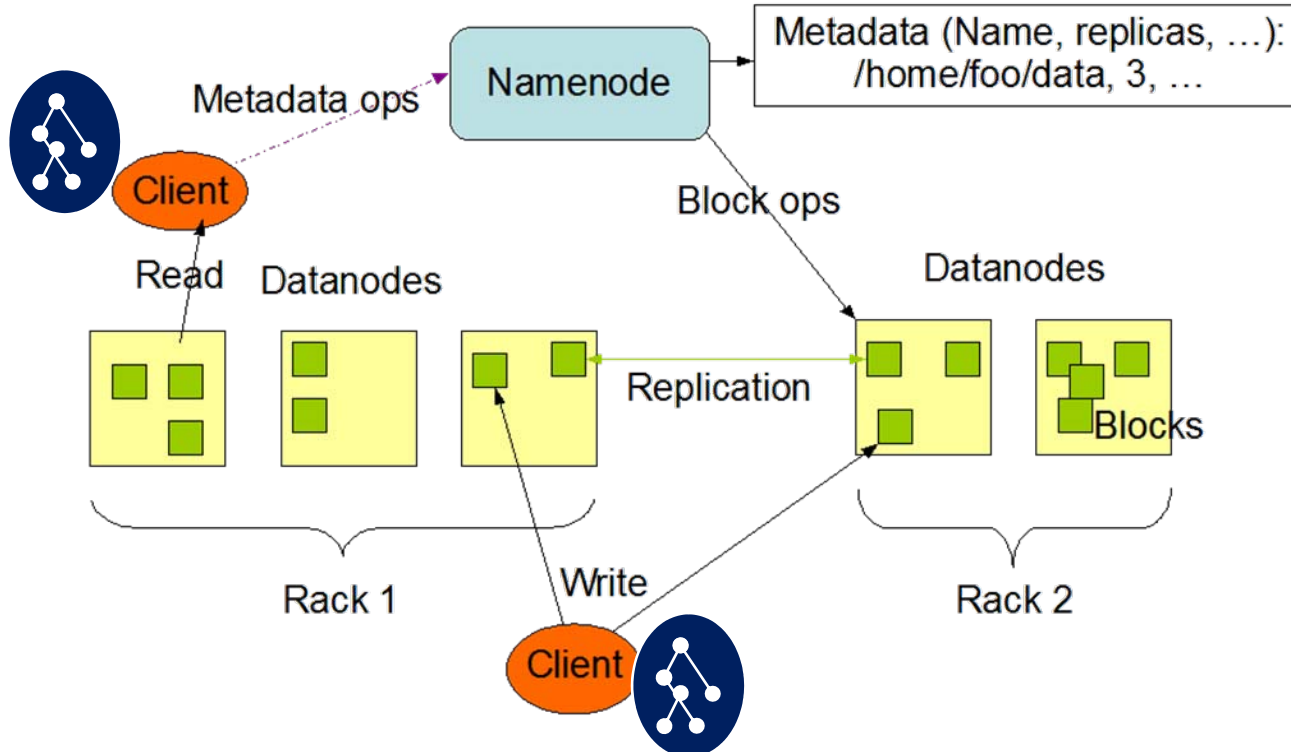


- Internal deployment but used by WW end-users
- Proprietary approach
- Asymmetric philosophy
- Thousands of chunk servers with 64MB chunk size (stripe unite)
- [research.google.com/pubs/papers.html](http://research.google.com/pubs/papers.html)



- GFS v2
  - ◆ Chunk size = 1MB !!
  - ◆ Hundreds of Distributed Masters (100 millions files managed by 1 master)

- Apache project
- Highly fault-tolerant built in Java
- Large data sets
- Asymmetric philosophy
- Files striped across DataNodes
- [hadoop.apache.org](http://hadoop.apache.org)



## ➤ Lustre

- ◆ Open source object-based storage system (based on NASD study from Carnegie Mellon Univ.)
- ◆ Asymmetric Philosophy
- ◆ Notion of Object (OST/OSS)
- ◆ [www.lustre.org](http://www.lustre.org)

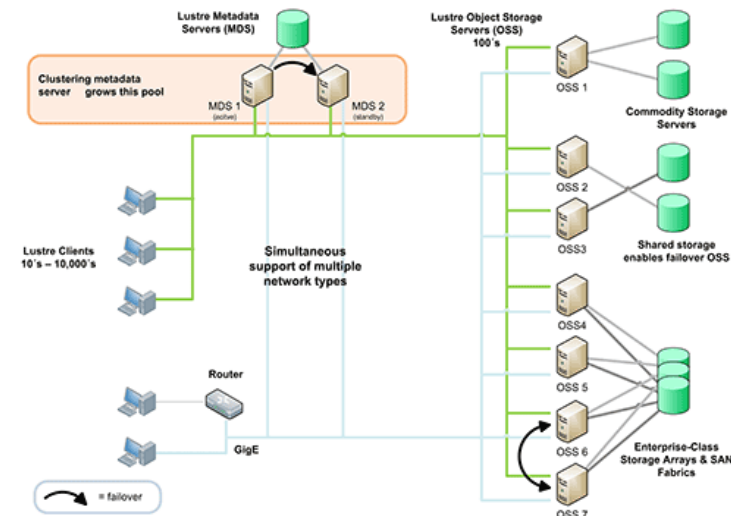
## ➤ PVFS (Parallel Virtual File System) now in 2<sup>nd</sup> gen.

- ◆ Project from Clemson Univ. and Argonne National Lab.
- ◆ Open source and based on Linux
- ◆ Asymmetric philosophy
- ◆ [www.pvfs.org](http://www.pvfs.org)

## ➤ MogileFS

- ◆ Application-level Distributed FS (no kernel module and no POSIX compliant), Asymmetric
- ◆ Open-source and Local filesystem agnostic
- ◆ Fully redundant, automatic file replication
- ◆ Flat Namespace and Shared-Nothing
- ◆ [www.danga.com/mogilefs](http://www.danga.com/mogilefs)

## ➤ Haystack, FineFS, again pNFS\* and CAS\* implementation



\* pNFS: Parallel NFS with NFS v4.1 | CAS: Content Addressable Storage

**Cloud Storage is simply the delivery of virtualized storage on demand.**

**SNIA proposes**

***Data Storage as a Service (DaaS),***  
**which means *delivery over a network of appropriately configured virtual storage and related data services, based on a request for a given service level.***

- Cloud Storage Initiative ([www.snia.org/cloud](http://www.snia.org/cloud))
- Cloud Data Management Interface (CDMI)
- Cloud Storage Use Cases and Reference Model
- White Paper «Cloud Storage for Cloud Computing »

# Cloud Storage Models

<p>Only Storage Cloud          (cloud-attached storage)</p>	
<p>Application + Storage          Same Cloud</p>	
<p>Application + Storage (primary          Cloud) + Secondary Storage for          Backup, Archiving or DR          (Secondary Cloud)</p>	
<p>Application Cloud connected to          Storage Cloud</p>	

## Notion of Private or Public Cloud

## ➤ Application

- ◆ Online & Primary File Storage, Online Backup and Data Archiving, DR... A new tier of Storage
  - Characteristics: Thousands/millions concurrent IOPS, multiple data/file format
  - Needs: High resiliency and geo data distribution, security and data privacy, performance and capacity scalability, open data access protocols or open API, contract based (SaaS)
  - Options: embedded data protection, data distributed across file/storage servers

## ➤ Configuration

- ◆ Distributed and Scalable File System, Cluster File System
- ◆ FUSE (Filesystem in USER space)
- ◆ VLFS as the industry introduces VLDB
  - ◆ File System -> File Manager -> File Storage
- ◆ “Everything is a File”

# Basic & Advanced File Services

## Basic

- Global Namespace (File Virtualization)
  - ◆ Organize storage in an overlay namespace
- Migration
  - ◆ Move files from one server to another
- Tiering / ILM
  - ◆ Move files via policy to the “best” storage
- Load Balancing
  - ◆ Move files to better distribute capacity or load
- Data Protection
  - ◆ Snapshot to support online data protection
  - ◆ Replication as a BC strategy
- Reporting & Statistics

## Advanced

- HA/BC/DR\*
  - ◆ Associated with data replication
- Data Classification and Optimized Placement
  - ◆ Data value and storage characteristics alignment
- Storage Capacity Optimization
  - ◆ DeDuplication, Reduction and Compression
- Quota Management
  - ◆ Report and enforcement
- Content Indexing & Search
  - ◆ File content to optimize file placement
  - ◆ Search for legal discovery...
- Application Acceleration
  - ◆ Local and Distributed file access
  - ◆ Notion of Distributed ILM/Tiering
- Security
  - ◆ Access Control, auditing and Encryption...

\* High-Availability/Business Continuity/Disaster Recovery

# Applications/File Storage Matrix

	HPC	DB OLTP	Web farm	Office Env.	Multimedia Video	Data Acquisition	Archiving	Cloud Storage	ILM	Other Vertical
<b>CFS</b>	(x)	x	x		x	x		x	x	x
<b>SANFS</b>	x		(x)		x	x				x
<b>NAS (+ FAN)</b>	x	x	x	x	x	x	x	x	x	x
<b>NAS Cluster</b>	x	x	x		x	x		x		x
<b>Distributed (&amp; Parallel)</b>	x				x	x	x	x		x
<b>WAFS (WAN Opt. &amp; Acc.)</b>				x						x
<b>CAS/Worm (with NAS access)</b>							x		x	

# Conclusion

## ➤ File System and File Storage technologies are key for current and next IT and Information, Data and Storage Challenges

- ◆ Scalability is a multi-dimension metric (Performance, Capacity, Availability, Manageability)
- ◆ Asymmetric (Master/Slave) seems to be a more scalable philosophy
- ◆ Unified, Global Namespace is fundamental
- ◆ File/Object approach is superior (pragmatic) on block approach
- ◆ Reliability & Security are key (authorization, authentication, privacy...)
- ◆ Think standard (de facto and industry) as many of them evolve (pNFS)
- ◆ Commodity hardware is more and more common (reality)



## ➤ Many approaches and philosophies in the industry

- ◆ There is no single solution that is superior in all cases BUT these approaches deliver real applications and business benefits for different applications needs
- ◆ Study and choose the one which delivers the best value for you

➤ Please check also the following tutorials



Check out SNIA Tutorial:  
**The File Systems Evolution**



Check out SNIA Tutorial:  
**Object-Based File Systems: an overview**



Check out SNIA Tutorial:  
**Global Namespaces for Summer**

- Please send any questions or comments on this presentation to SNIA
  - ◆ [trackfilemgmt@snia.org](mailto:trackfilemgmt@snia.org)  
(File Systems & File Management)

**Many thanks to the following individuals  
for their contributions to this tutorial.**

**- SNIA Education Committee**

**Philippe Nicolas**



Education

# **Find and Select the Right File Storage for your Applications**

Philippe Nicolas, KerStor